63-2-4

TM 970 001 00

A [illegible] P[illegible] [illegible]OVAL [illegible]

TM-970/001/00

TECHNICAL MEMORANDUM

(TM Series)

This document was produced in connection with a research project sponsored by SDC's independent research program.

A One Pass JOVIAL Compiler

by

E. Book
H. Bratman
J. Schwartz

22 January 1963

SYSTEM DEVELOPMENT CORPORATION
2500 COLORADO AVE.
SANTA MONICA
CALIFORNIA

A ONE PASS JOVIAL COMPILER

An investigation into improved techniques in compiling and debugging has resulted in the production of a small fast compiler which operates on a subset of JOVIAL. This subset has been named JOVIAL-X.2. The compiler processes a program on the IBM 7090 and produces 7090 binary instructions. The compiler was programmed and checked out in four months by E. Book, H. Bratman, and J. Schwartz.

Design and Programming Considerations

The compiler can be roughly divided into three segments--a generator, a translator and common routines. The generator does the work of the two passes of the J-3 generator, but because of a complete redesign (and partially because it is processing a simpler language) it is smaller and more efficient. The translator is similar in design to J-3 translators. It is smaller principally because the language is simpler.

The generator analyzes one source statement and produces an intermediate language representation of the statement. The translator then produces the necessary binary instructions, assigns locations, and stores the instruction in an output buffer. The generator while processing an item declaration produces a dictionary entry, assigns locations and, if necessary, stores data constants into the output buffer. This process is repeated statement by statement. When the buffer is filled the binary instructions are written onto a tape. Locations that are unknown at the time of instruction generation are filled in later via subsequent "correction records" on the tape. The binary tape is provided with a self-loading loader which reads in the binary instructions and correction records, fills in the missing addresses, and then passes control to the program.

The compiler was written in JOVIAL-X.2 and was first compiled via the Q-32 J-2 JOVIAL compiler. When the compiler was running fairly successfully (after 8 weeks), compilation and checkout was shifted to the 7090 and its JOVIAL compiler. Eight more weeks were necessary to complete checkout on the 7090. The Q-32 was used first because of its availability and accessability during the difficult early phases of checkout. Its associated debugging routines were extremely useful at this stage. Fortunately the 7090 was also readily available when it was needed. Approximately 12 hours of Q-32 time and 11 hours of 7090 time were used to complete the first phase of checkout -- which was self compilation of the one pass compiler.

There were at least four good reasons for the rapid completion of the first phase -- 1) The experience of the programmers in this area. 2) In many respects the subset of JOVIAL used made the compilation process almost trivial. 3) The ready availability of computer time. 4) The speed of the one pass compiler greatly speeded up the final stages of the checkout process.

Compiler Performance

The compiler at present consists of approximately 13,500 operating instructions and 13,400 words of data, tables, items and constants.

The object program can optionally be executed in a "compile and go" mode, or it can be executed repeatedly from a binary tape.

The compiler produces instructions at the rate of approximately 2600 binary instructions per minute, simultaneously producing an interspersed source and assembly type listing. Eliminating the assembly type listing results in a compilation rate of approximately 5000 binary instructions per minute. Compared with the current 7090 J-2 compiler; these rates are six times faster with the assembly listing and ten times faster without.

By comparing the compiler produced via the J-2 compiler and the one produced via a self-compilation a few tentative statements can be made concerning relative efficiency of the compiled program. A program compiled by the one-pass compiler is approximately 20% longer and 20% slower than a program compiled by the 7090 J-2 compiler.

Future Plans

The completion of the first phase of this compiler, that is its capability to compile itself leaves a number of possibilities open for future development. Essentially this compiler is a tool which may be sharpened, it can be used as a prototype to produce similar tools which are superior, and it can be extended to make it adequate for systems work.

First, sharpening the tools, here the compiler is polished so that it runs faster, produces more efficient object programs, and those features of the language not checked out by self compilation are improved. Convenience features are added both to the language and the compiler. Some of these would be library, automatic formatting, debugging statements etc.

Second, another compiler could be produced using this compiler as a prototype. This would involve investigating radically different designs in search of new compiling techniques. Such areas as statement decomposition algorithms, translation algorithms, systemization of compiler structure using a syntax directed or table directed approach and various index register assignment algorithms. Changes to the language which would make it more useful or applicable to a wider range of problems or possibly permit efficiencies in the compiler and programs which it produces.

Third, extensions toward system languages and programming. The language processed is a subset of JOVIAL. Features of JOVIAL omitted from the first version of the compiler could be reincorporated. By using some of the techniques developed all the JOVIAL compilers might be improved.

These three courses of action are neither mutally exclusive nor jointly exhaustive. However simultaneous investigation along the three indicated paths would be possible if people were available.

Documentation

Further documentation is being prepared that will describe in greater detail the compiler design, the language, and operating instructions.

UNCLASSIFIED

System Development Corporation,
Santa Monica, California
Scientific rept., TM-970/001/00, by
E. Book, H. Bratman, J. Schwartz.
22 January 1963, 3p.
Unclassified report

DESCRIPTORS: Digital Computers.
Machine Translation.

Identifiers: JOVIAL.

Describes the performance of the one pass JOVIAL compiler which processes

UNCLASSIFIED

UNCLASSIFIED

a program on the IBM 7090 computer and produces binary instructions. The compiler operates on a subset of the JOVIAL language. It compiles ten times faster than the present 7090 JOVIAL compiler.

UNCLASSIFIED